

Практична робота №1. Основи роботи з Node-RED

Node-RED - це інструмент програмування для об'єднання апаратних пристроїв, API та онлайн-послуг новими та цікавими способами.

Редактор функціонує на основі браузера, а програма виглядає як об'єднані в потоки (flow) вузли (node), що вибираються з палітри.

Після редагування потоки можуть бути розгорнуті в середовище виконання в один клік. Вбудована бібліотека дозволяє зберігати корисні функції, шаблони або потоки для повторного використання.

Легке середовище виконання побудоване на Node.js, що використовує подіємо-керовану неблокуючу модель. Це робить його ідеальним для роботи на краю (Edge) мережі на недорогих апаратних засобах, таких як Raspberry PI, а також у хмарі.

З більш ніж 225 000 модулів у репозиторію пакетів Node, легко збільшити діапазон вузлів палітри для додавання нових можливостей.

Потоки, створені в Node-RED, зберігаються за допомогою JSON, які можна легко імпортувати та експортувати для спільного використання з іншими.

Онлайн бібліотека потоків дозволяє вам поділитися своїми найкращими потоками зі світом.

Про Node.JS

[wiki](#)

1. Інсталяція Node-RED під Windows

1.1. Завантаження Node.JS

Завантажити [msi-файл](#) Node.JS LTS версії

1.2. Встановлення Node.JS

Запустити на виконання msi-файл від імені адміністратора і встановити Node.JS, при виклику діалогових вікон все залишати за замовченням

1.3. Визначення версії npm

Після інсталяції запустіть командний рядок (CMD), у якому введіть

```
node --version && npm --version
```

повинно вивести версію node() - npm ()

npm ([Node Package Manager](#)) - це менеджер пакунків для мови програмування JavaScript. Для середовища виконання Node.js є менеджером пакунків за замовчуванням. Включає в себе клієнт командного рядка, який також називається npm, а також онлайн-базу даних публічних та приватних пакунків, яка називається реєстром npm. Реєстр доступний через клієнт, а доступні пакунки можна переглядати та шукати через веб-сайт npm. Менеджер пакунків та реєстр керуються npm, Inc.

1.4. Інсталяція Node-RED

Інсталюйте Node-RED . Інсталювання проводиться з використанням npm з командою install.

Наберіть в командному рядку

```
npm install -g --unsafe-perm node-red
```

почнеться процедура інсталяції

1.5. Запуск Node-RED.

Запустіть Node-RED з командного рядка

```
node-red
```

Можуть бути показані повідомлення про пропозицію розблокування брандмауером, з якими треба погодитись.

1.6. Відкриття редактору.

Відкрийте браузер, перейдіть до редактору Node-Red, за посиланням <http://127.0.0.1:1880/>. Для того, щоб Node-RED виконувався, вікно з командним рядком не можна закривати.

2. Знайомство з Node-RED

Програма створена на Node-RED складається з **потоків (Flow)**, які виконуються як умовно незалежні програми. Потоки – це зв'язані між собою інформаційними **дротами (wires)** **вузли (Node)**, що виконують певну функцію. Таким чином, ідеологія програмування Node-RED дещо схоже на побудову програм на мові FBD, що є стандартною для програмування ПЛК (IEC 61131-3). Тим не менше, між цими мовами є значні відмінності.

2.1. Ознайомлення з редактором Node-RED.

Відкрийте в браузері редактор Node-RED, якщо він ще не відкритий. Ознайомтеся з його основними частинами (рис.1)

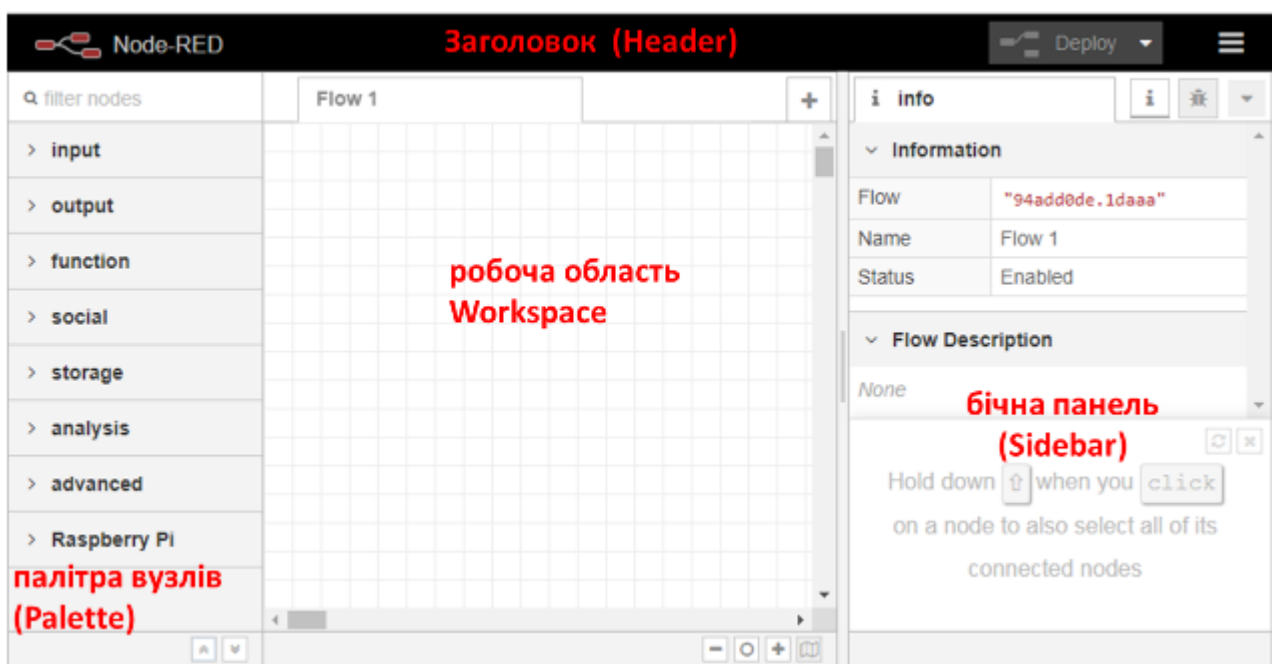


рис.1. Вигляд редактору Node-RED

2.2. Розміщення вузлів inject і Debug

Виберіть з палітри і розмістіть на робочій області вузли

- **Input->inject**
- **Output->Debug**

З'єднайте їх між собою. Повинно вийти як на рис.2. Блакитні кружечки значать, що зміна в вузлах ще не відобразилася в середовищі виконання, так як змінена програма не була в ньому розгорнута.

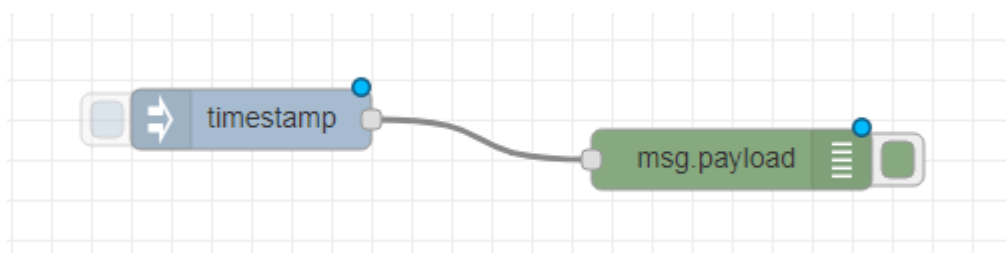


рис.2. Вигляд програми

2.3. Розгортання

В заголовку виберіть пункт **Deploy->Modified Nodes** (див.рис.3), після чого натисніть **Deploy** (Розгортання).

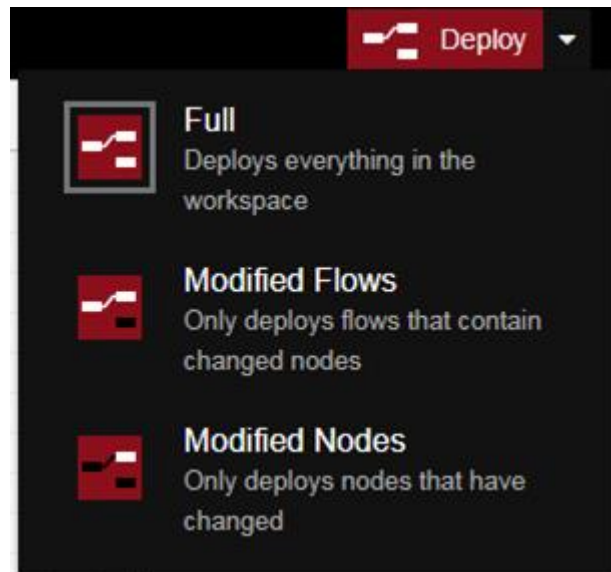


рис.3. Вибір варіанту розгортання

При вдалому розгортанні з'явиться повідомлення.

А в робочому просторі вузли вже будуть без блакитних кружечків.

2.4. Відображення вікна повідомлень

Для перевірки роботи програми, на бічній панелі треба відобразити вікно **Debug messages** (налагоджувальні повідомлення) шляхом натиснення кнопки з «жуком».

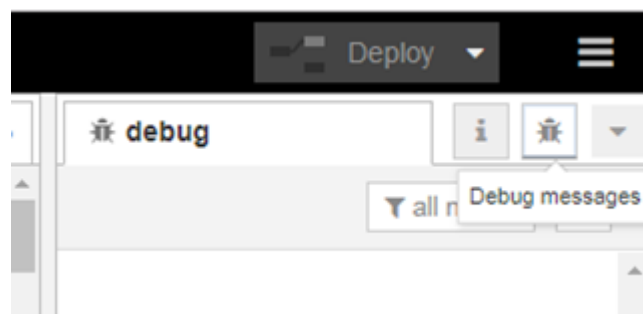


рис.4. Показати вікно повідомлень

Ліворуч вузла типу Inject з назвою «timestamp» є кнопка, яка приводить до ініціювання розрахунку ланцюжка вузлів, що починаються з нього. Натисніть на цю кнопку.

В результаті з'явиться повідомлення про успішне вприскування (Inject) а на панелі повідомлень з'явиться повідомлення (рис.5).

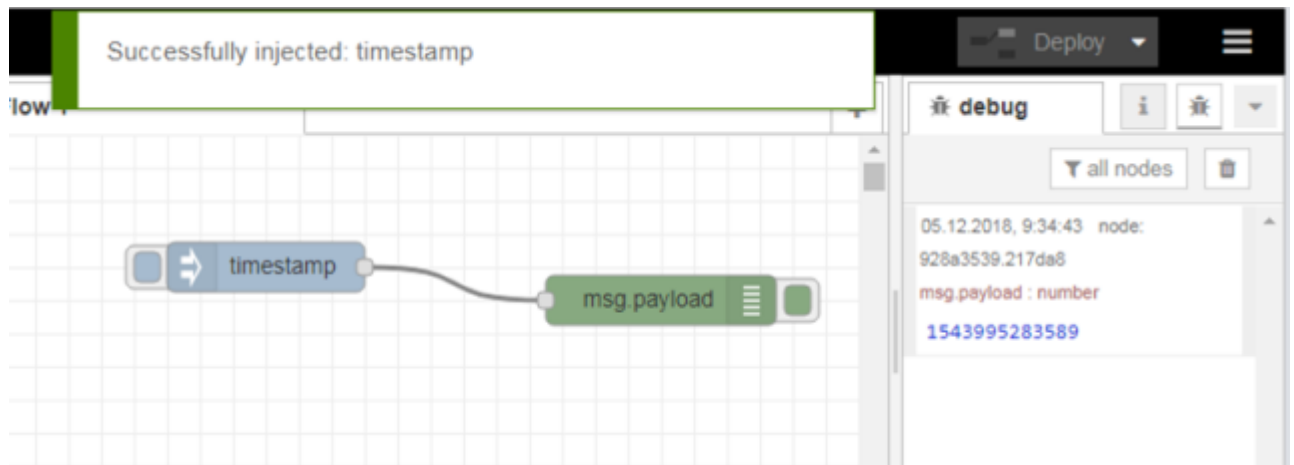


рис.5. Результат виконання програми

На цьому прикладі розглянемо, як виконується програма.

У більшості випадків перерахунок вузлів починається тоді, коли на його вхід подається повідомлення (message). Повідомлення – це прості об’єкти (типу структурні змінні) JavaScript що можуть мати будь який набір властивостей. Тобто в даній програмі після перерахунку вузла з іменем «timestamp» буде сформовано об’єкт-повідомлення (**msg**) і переданий по дроту вузлу з іменем (msg.payload).

На вхід вузла з іменем «timestamp» повідомлення не надходять, бо він є ініціатором розрахунку. Всі вузли палітри що входять в групу **Input** є ініціаторами розрахунку. Ініціація вузлів типу **Inject** відбувається шляхом ручного запуску по кнопці, або через певні інтервали часу, що налаштовується у вузлі. Ініціювання повідомлення це формування полів **msg** та відправка його іншим вузлам по дротам. Повідомлення, надіслане вузлом Inject, має властивості **payload** (корисне навантаження) та **topic** (тема). За замовченням Inject записує у властивість topic відмітку часу (timestamp – кількість мілісекунд з 1980 року).

Вузол типу Debug «msg.payload» використовується для відображення повідомлень на бічній панелі Debug. Таким чином, після отримання повідомлення, цей вузол відображає його зміст на бічній панелі.

2.5. Налаштування Inject на періодичне оновлення

Змініть налаштування властивостей вузлів, як показано на рис.6: змініть імена вузлів, вкажіть тему (topic) та періодичність оновлення для вузла типу Inject. Вікно налаштування з’являється по подвійному кліку по вузлу.

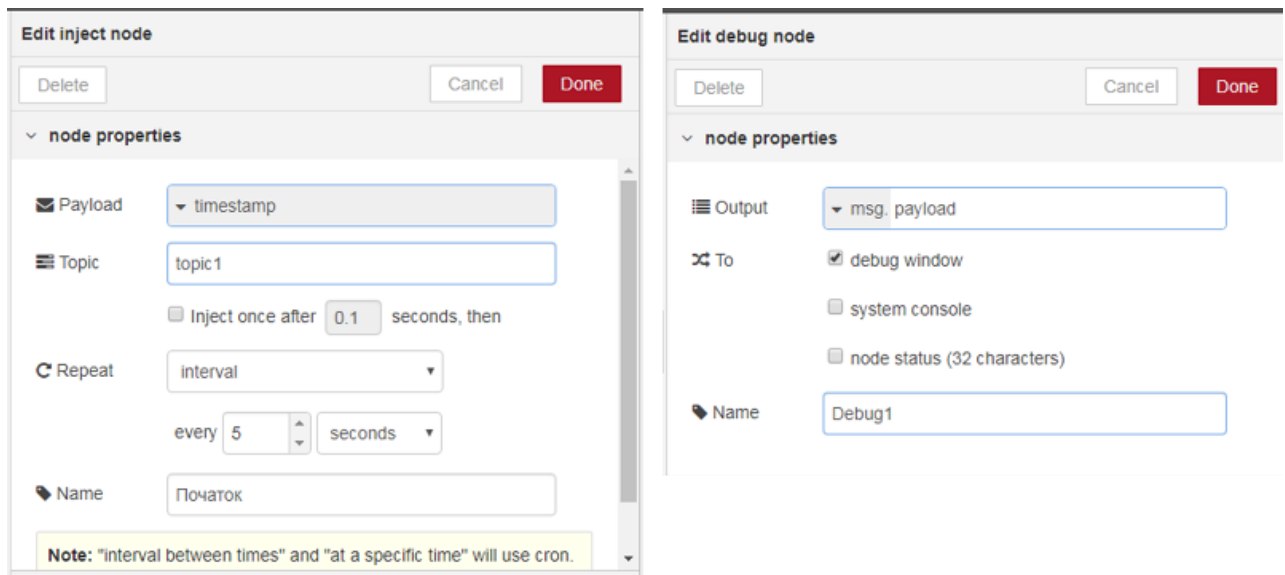


рис.6. Налаштування вузлів

Зробіть розгортання, та проаналізуйте зміст виведених у вікні Debug повідомлень.

2.6. Налаштування Inject на текстове повідомлення

Змініть вузол «Початок» так, щоб він формував корисне навантаження текстом «Це текстове повідомлення» (рис.7), та проаналізуйте як воно виводиться на вікно Debug.

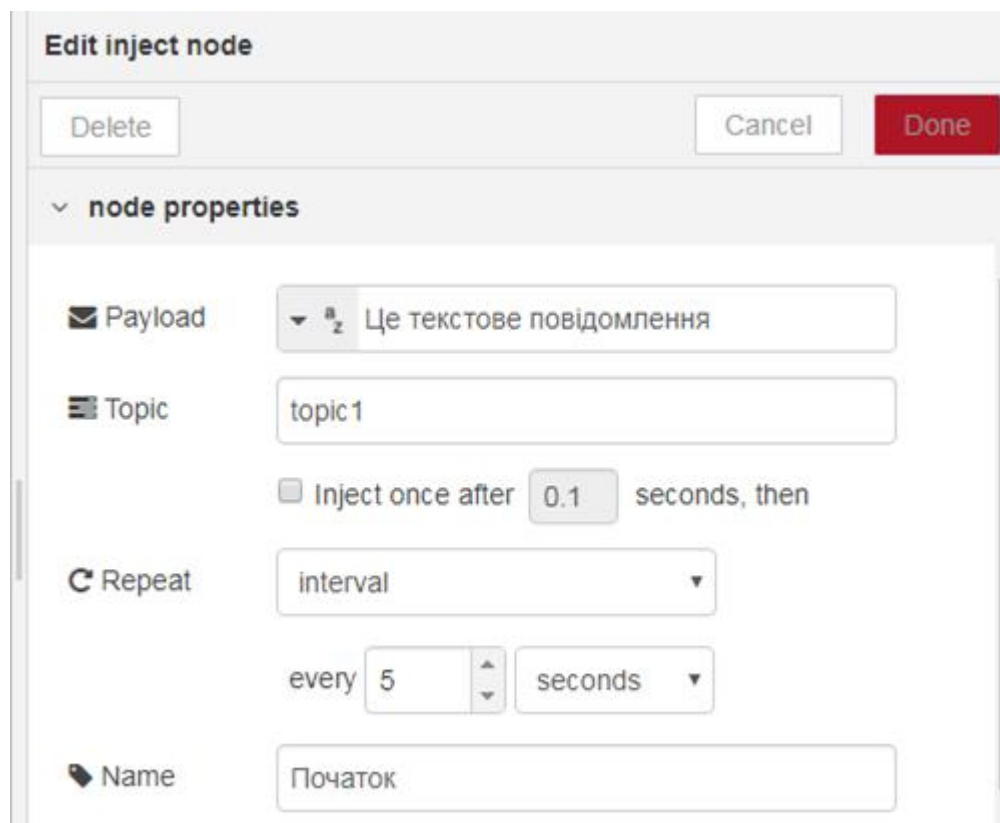


рис.7. Формування текстового повідомлення

2.7. Використання вузлів типу change та delay

Змініть вузол «Початок» так, щоб він знову формував корисне навантаження відміткою часу (Timestamp). Розгорніть (deploy) програму та проконтролюйте щоб відмітка часу кожні 5 секунд відображалася у вікні повідомлень

Ознайомтеся з роботою вузлів типу **change** та **delay** в інструкції користувача.

Змініть програму, як показано на рис.8, використовуючи вузли delay (“delay 1s”...”delay 4s”) та change (“set1”...”set5”). Для вузлів delay виставте затримки:

- delay 1s – 1 seconds
- delay 2s – 2 seconds
- delay 3s – 3 seconds
- delay 4s – 4 seconds

Для вузлів change виставте правило рівним «set», та змініть властивості «to» на наступні текстові поля:

- set1 – один
- set2 – два
- set3 – три
- set4 – чотири
- set5 – п'ять

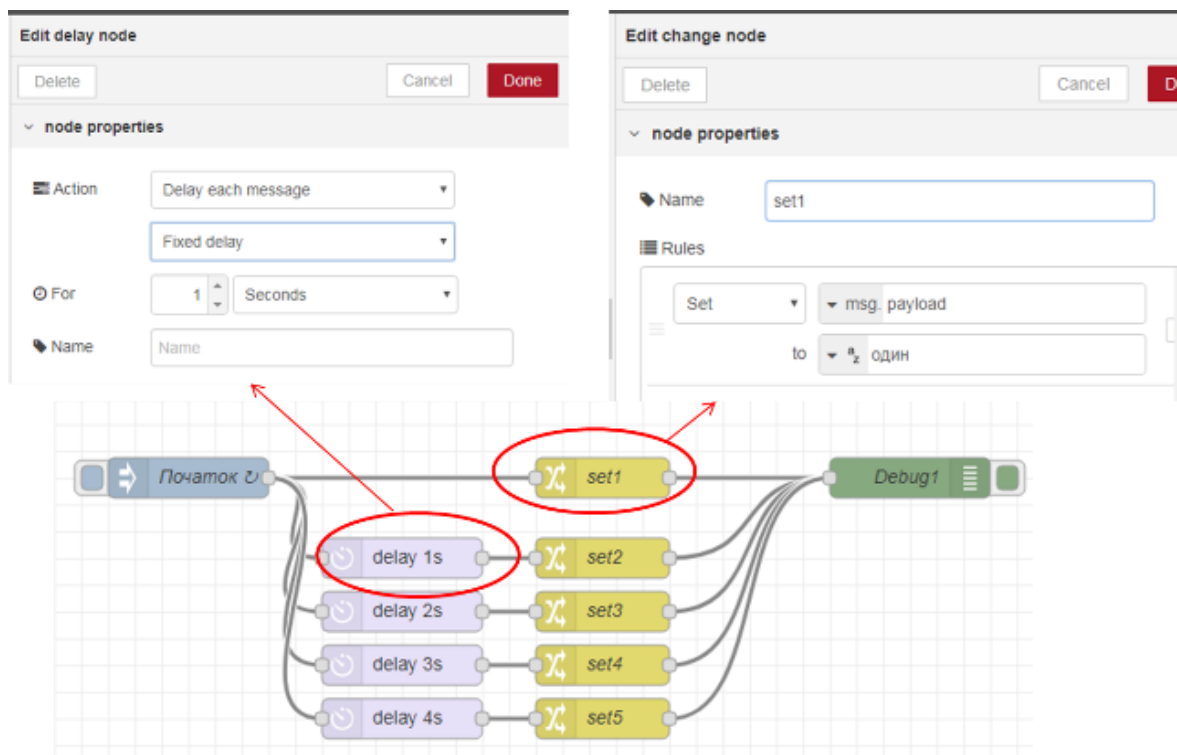


рис.8. Використання вузлів типу change та delay

Розгорніть (deploy) програму та проконтролюйте щоб кожної секунди у вікні повідомлень виводилося конкретне повідомлення від «один» до «п'ять».

2.8. Ознайомлення з роботою вузлів типу function

Ознайомтеся з роботою вузлів типу function з інструкції користувача

Вузол function може обробляти повідомлення з використанням javascript. Змініть програму так, щоб відмітка часу виводилася в форматі дати та часу. Для цього використовується об'єкт типу Date та його метод toLocaleString(). Зробіть розгортання та переведіть вузол “Debug1” в режим приховання повідомлень.

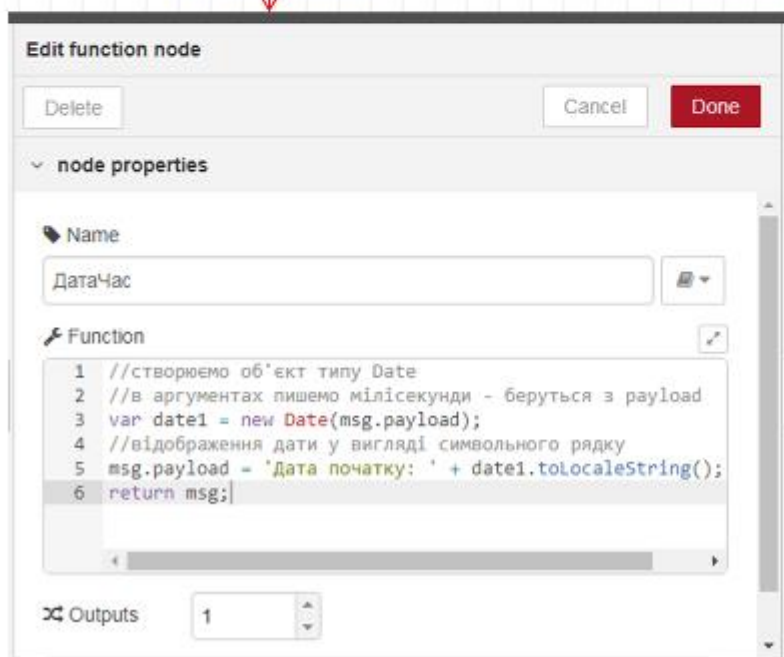
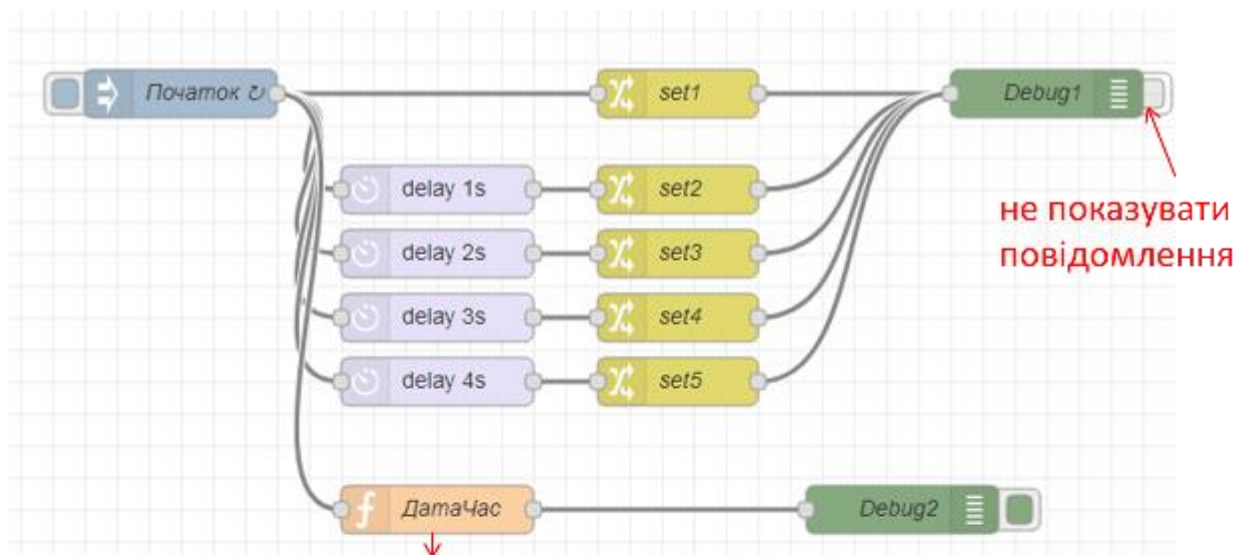


рис.9. Змінена програма з вузлом function

Про синтаксис **javascript** можна дізнатися [тут](#) (укр. мова)

Про об'єкт **Date** можна прочитати [тут](#)

3. Підключення та ознайомлення з модулем node-red-dashboard

Node-RED надає можливості розробки WEB-інтерфейсу користувача. Це робиться за допомогою модуля node-red-dashboard, який необхідно встановити. Додатково про node-red-dashboard можна прочитати у [довіднику](#).

Node-RED дозволяє інстальовати та оновлювати палітру вузлів. Це робиться через Manage Palette (рис.10) . Деталі інсталяції читайте в [інструкції користувача](#).

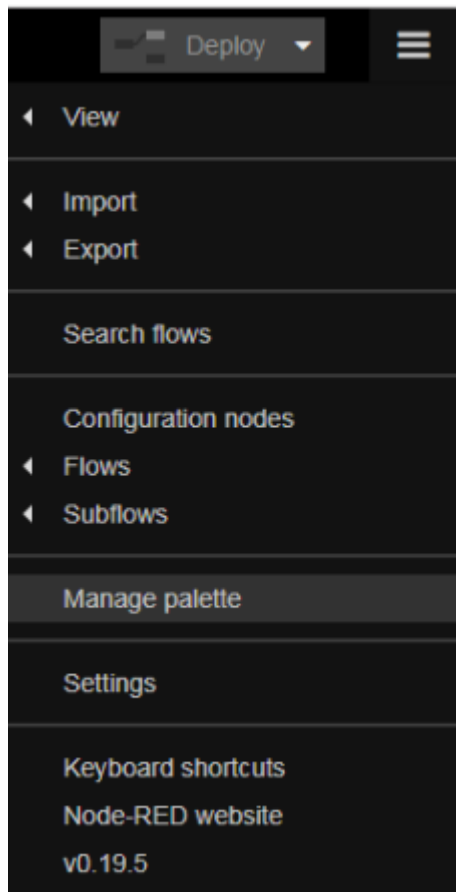


рис.10. Вибір Manage palette

3.1. Інсталяція node-red-dashboard

В налаштуваннях палітри на вкладці Install в поле фільтру введіть node-red-dashboard і інсталюйте даний пакет (рис.11):

- натисніть кнопку install
- підтвердьте інсталяцію у вікні повідомлення
- після інсталяції закрийте вікно керування палітрою

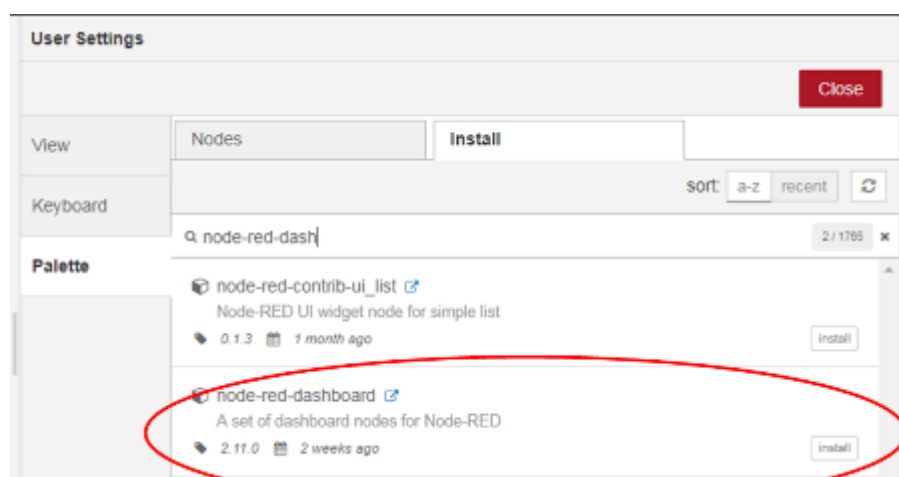


рис.11. Вибір Manage palette

Перевірте чи з'явилася в палітрі розділ Dashboard

3.2. Додавання закладок

Після встановлення у бічній панелі з'явилася нова іконка з зображенням діаграми (рис.12) . Натисніть на ній.

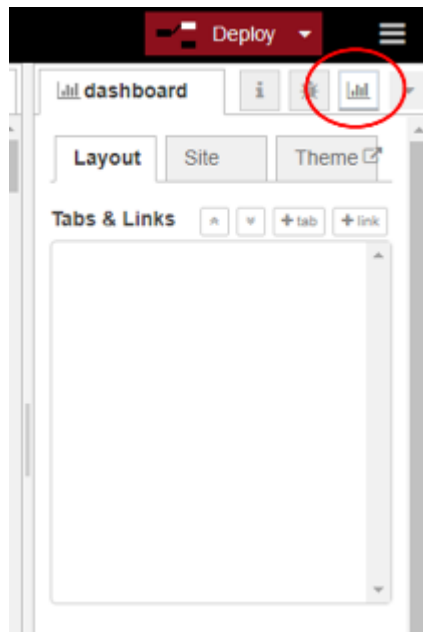


рис.12.

В Layout додайте дві закладки (tab) та змініть їх назви як це показано на рис.13.

Ім'я першої закладки повинно називатися Вашим прізвищем та ім'ям, наприклад «Іваненко Іван».

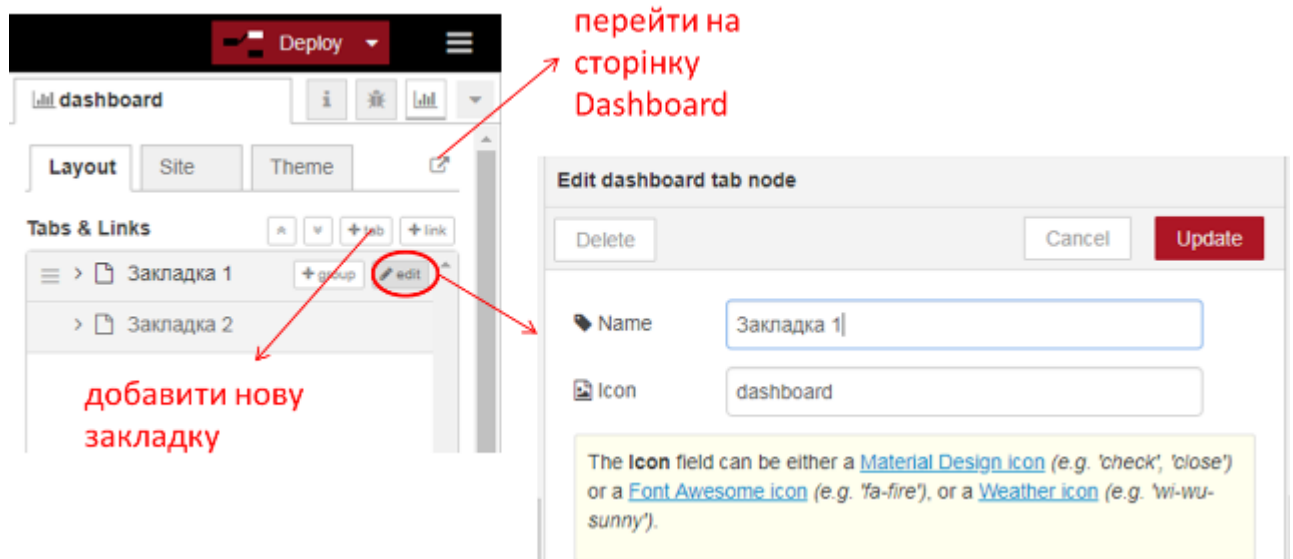


рис.13

3.3. Створення вузлу Dashboard text для виведення дати та часу

Модифікуйте програму, створивши вузол типу dashboard-> text і підєднавши його до вузла «ДатаЧас» (див. рис.14). Налаштуйте вузол відповідно до рис.14 однак ім'я першої групи повинно називатися Вашим прізвищем та ім'ям «Іваненко». Створіть ще одну групу, яка повинна називатися Вашим ім'ям «наприклад Іван».

Після усіх налаштувань зробить розгортання, відкрийте створений Dashboard, шляхом натискання кнопки переходу (див.рис.13.), або ввівши в новій вкладці браузера <http://127.0.0.1:1880/ui>

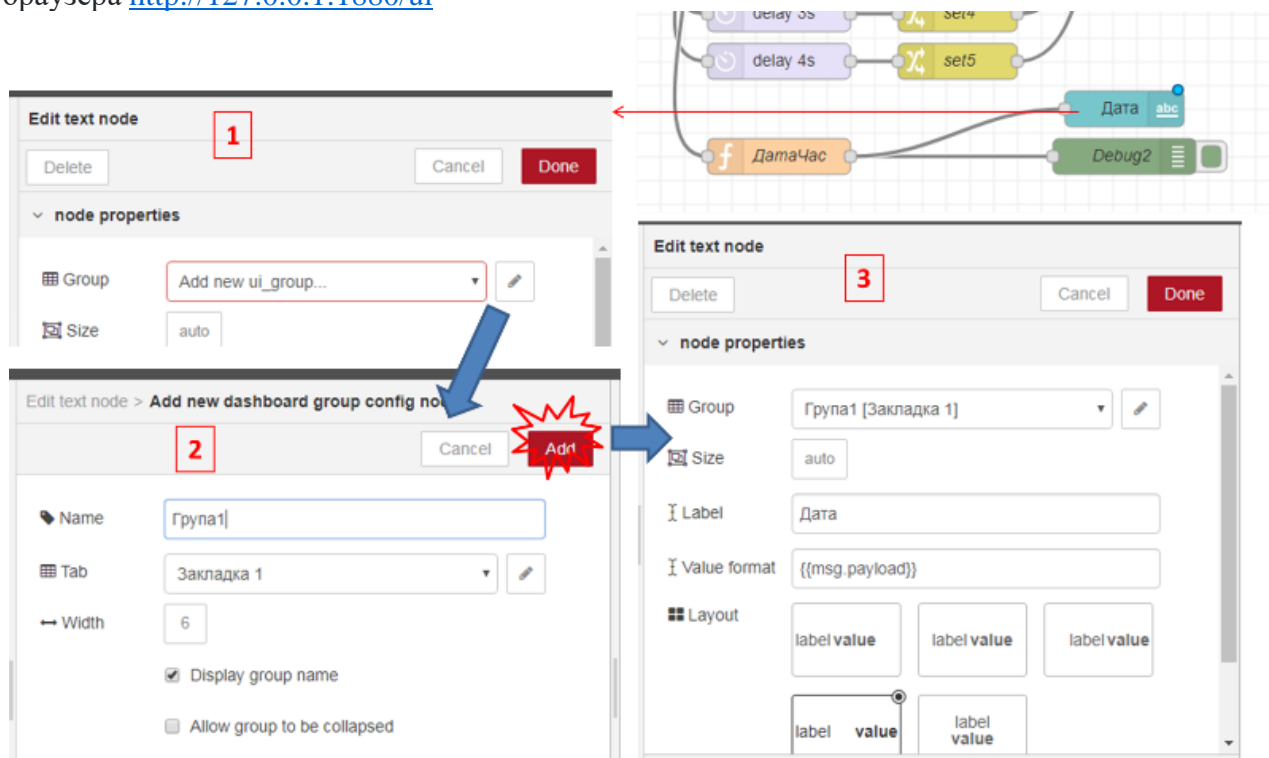


рис.14.

На вкладці повинно з'явитися щось типу такого, як показано на рис.15

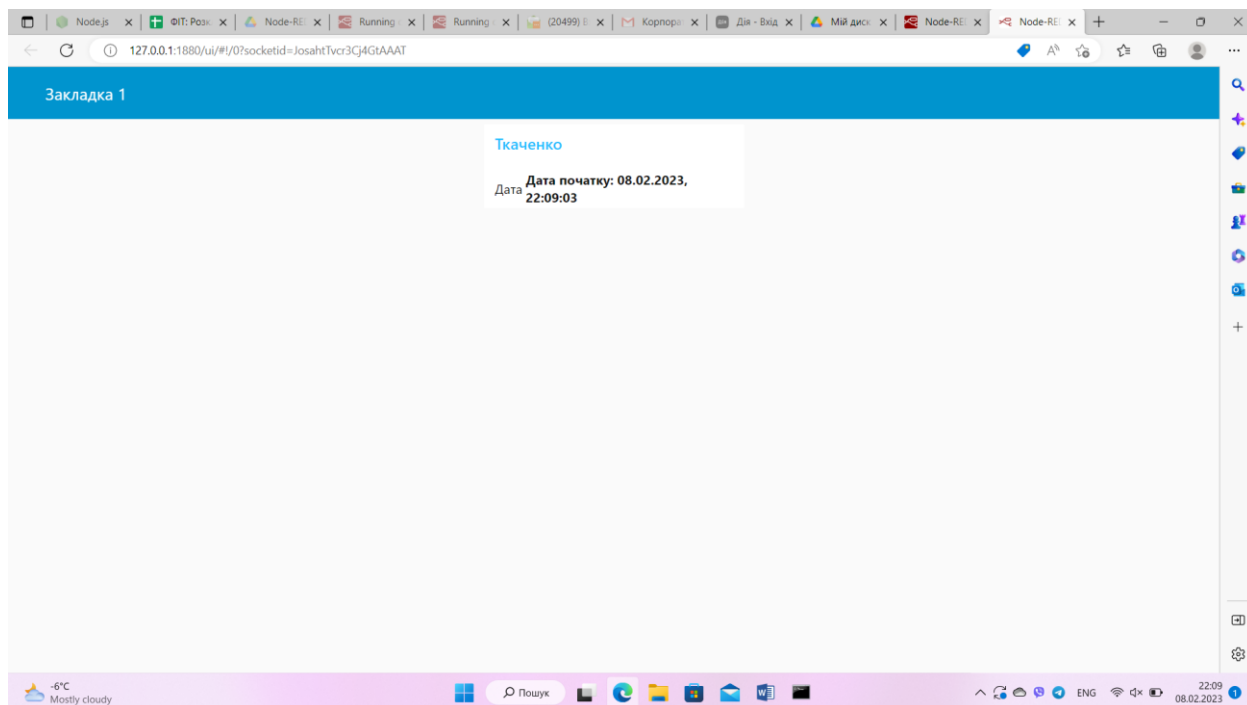


рис.15

3.4. Створення вузлу Dashboard text для виведення числа прописом
Аналогічним чином зробить для відображення числа прописом.

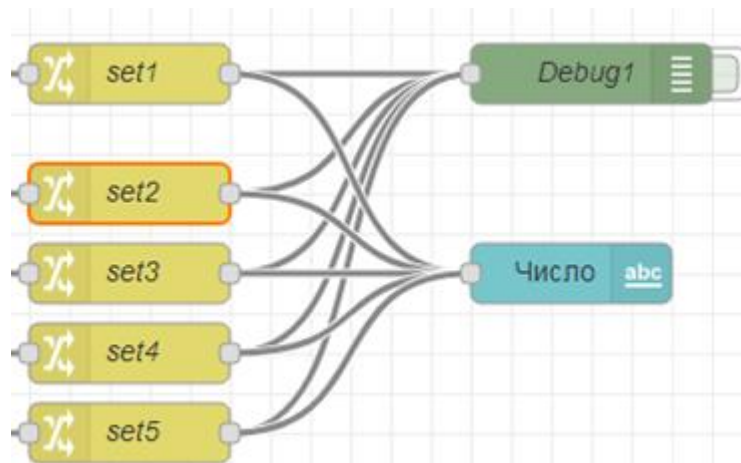


рис.16

3.5. Використання вузлів Slider, Gauge, Audio out

Добавте до програми фрагмент, як показано на рис.17.

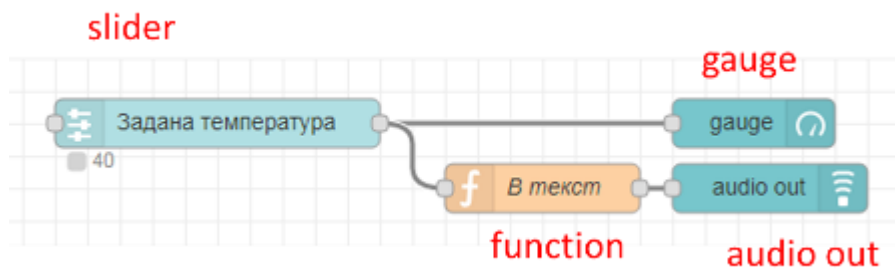


рис.17

Налаштування вузлів показані на рис.18-21

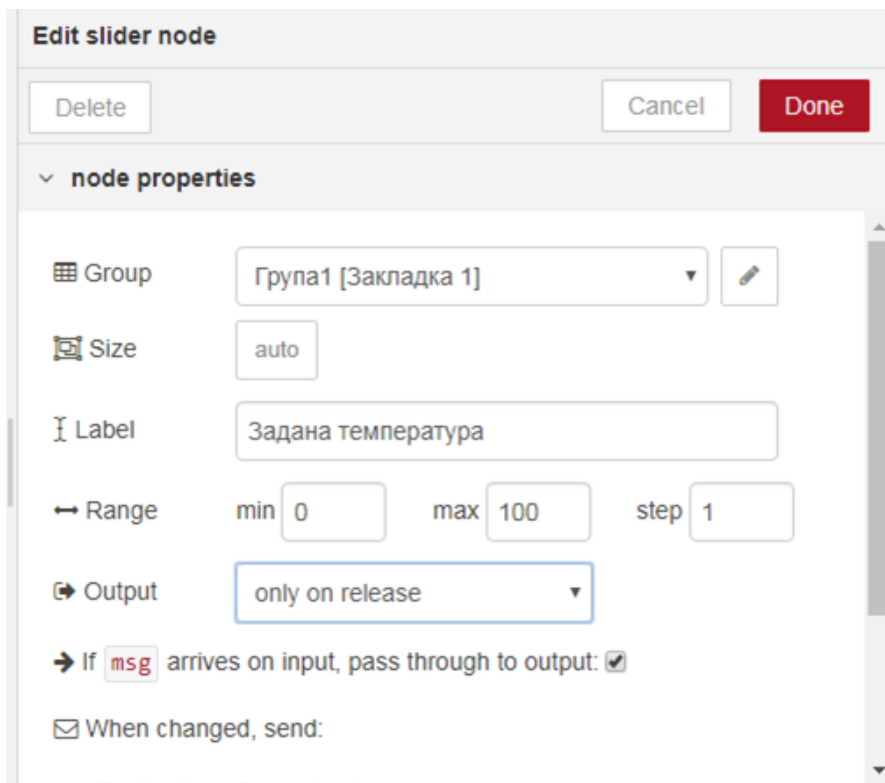


рис.18

Edit gauge node

Delete Cancel Done

▼ node properties

Size auto




Type Gauge ▼

Label gauge

Value format {{value}}

Units units

Range min 0 max 100

Colour gradient   

Sectors 0 ... 80 ... 90 ... 100

Name

рис.19

Edit function node

Delete Cancel Done

▼ node properties

Name В текст

Function

```

1 msg.payload = msg.payload.toString() ;
2 return msg;

```

Outputs 1

рис.20

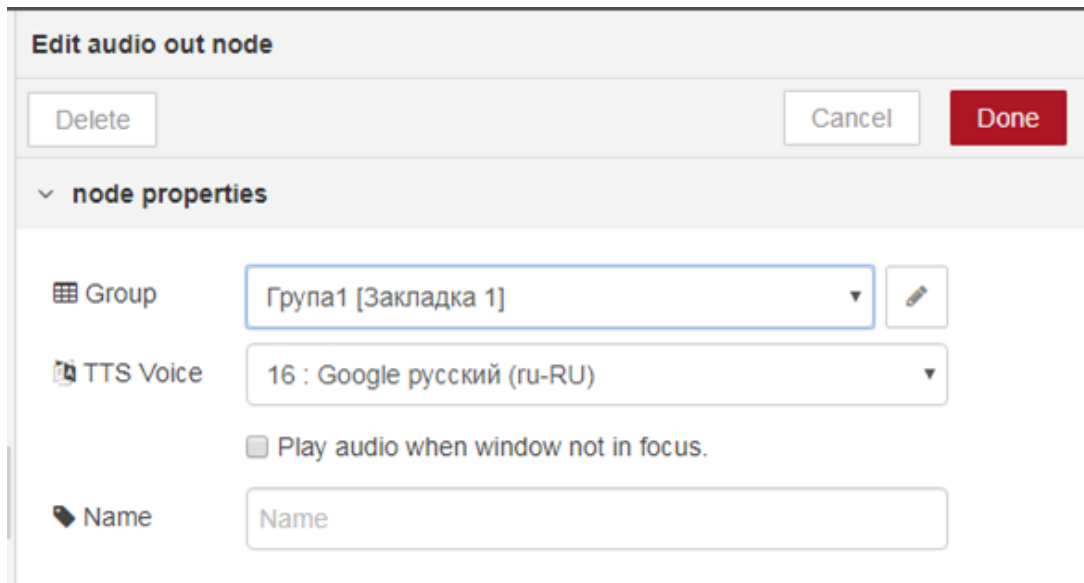


рис.21

Зробіть розгортання і подивіться результат на вікні Dashboard.

3.6. Робота з вузлом switch

Ознайомтеся з принципами роботи вузла **switch**. Модифікуйте програму відповідно до наведеної на рис.22. Вузли налаштуйте відповідно до рис.23-26.

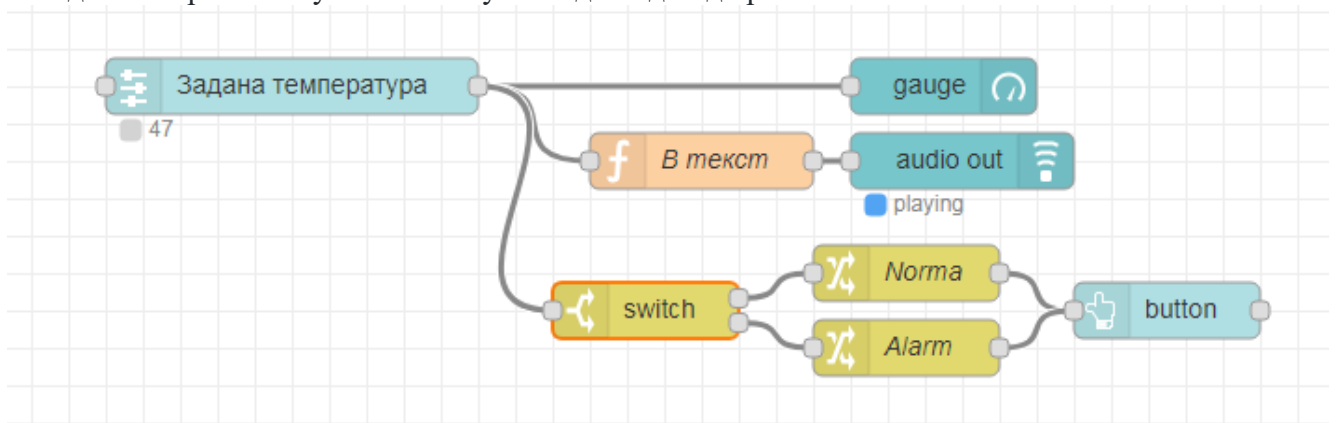


рис.22. Фрагмент модифікованої програми

Edit switch node

Delete Cancel Done

▼ node properties

Name

Property ▼ msg. payload

is between ▼ ▼ 0₉ 0 → 1

and ▼ 0₉ 50

>= ▼ ▼ a_z 51 → 2

рис.23. Налаштування вузла switch

Edit change node

Delete Cancel Done

▼ node properties

Name

Rules

Set ▼ ▼ msg. payload

to ▼ a_z HOPMA

Set ▼ ▼ msg. color

to ▼ a_z GREEN

рис.24. Налаштування вузла norma

Edit change node

Delete Cancel Done

▼ node properties

Name Alarm

Rules

- Set msg. payload to a2 АВАРІЯ
- Set msg. color to a2 RED

рис.25. Налаштування вузла Alarm

Edit button node

Delete Cancel Done

▼ node properties

Group [Закладка 1] Група1

Size auto

Icon optional icon

Label {{msg.payload}}

Tooltip optional tooltip

Colour optional text/icon color

Background {{msg.color}}

рис.26. Налаштування вузла Button

Зробіть розгортання проекту, перевірте як працює програма. Для цього на сторінці веб-інтерфейсу змініть значення заданої температури в діапазоні 0-50, а потім >50.

Ця частина програми працює наступним чином. При зміні значення температури, в msg.payload значення поступає на обробку в вузол switch, де на один із 2-х виходів формується повідомлення в залежності від тієї умови, яка спрацювала.

При виконанні умови $0 < \text{msg.payload} < 50$ (is between), повідомлення передається на перший вихід, до якого в свою чергу приєднаний вузол Norma (тип function->change). Той задає текстове значення для властивості msg.payload рівним «НОРМА» і формує нову властивість msg.color рівною «GREEN». Далі msg поступає вузол «button», який використовується для відображення тексту в прямокутнику. Значення тексту задається полі Label, а колір в полі Background. При формуванні динамічних значень для вузлів, використовується формат angular фільтрів, в якому вказується підстановка в подвійних фігурних дужках.

Аналогічна обробка проводиться при спрацюванні у вузлі switch умови $\text{msg.payload} > 50$. Повідомлення згенерується на другому виході, який активує перерахунок вузла «Alarm» що буде формувати текст та колір для кнопки.

Зробіть копії екранів програми та візуалізації для звіту до Вашої роботи.